

CAPITOLUL 16

INTRODUCERE IN PL/SQL

CE ESTE PL/SQL ?

- Un limbaj procedural
 - Un vehicul pentru procesarea tranzactiilor
 - O tehnologie disponibila in citeva aplicatii ORACLE
 - O extensie SQL
-
- Ce este PL/SQL ?
PL/SQL (Procedural Language /SQL) este o extensie la SQL, incorporand multe facilitati ale limbajelor de programare din ultimii ani. El permite ca manipularea datelor si procedurile de interogare din SQL sa fie incluse in blocuri stucturate si proceduri, PL/SQL devenind astfel un puternic limbaj de procesare al tranzactiilor.
 - Avantaje ale PL/SQL

1. Stuctura de bloc

PL/SQL este un limbaj structurat, adica programele pot fi impartite in blocuri logice, fiecare continind resursele necesare in acel bloc. variabilele pot fi declarate local in cadrul unui bloc in care vor fi folosite, iar tratarea erorilor (sau Exceptiile) se poate face in blocul in care apar.

2. Controlul executiei

Deciziile, bucelele si salturile pot fi folosite pentru a controla executia programelor, decizind daca si cind SQL si alte actiuni sa fie executate. Aceste facilitati permit aplicatiilor ORACLE cum ar fi SQL*Forms, de a grupa comenzi si de a controla executia lor prin PL/SQL.

3. Portabilitatea

Deoarece PL/SQL deriva din ORACLE, programele pot fo portate pe toate masinile ce suporta ORACLE si PL/SQL.

4. Integrarea

PL/SQL joaca un rol din ce in ce mai important in RDBMS si in aplicatiile ORACLE. Variabilele si tipurile de date din PL/SQL sunt compatibile cu cele din SQL. PL/SQL acopera golul dintre nevoia de access la tehnologia bazelor de date si nevoia de facilitati pentru programarea procedurala.

5. Performantele

Utilizarea PL/SQL poate ajuta la imbunatatirea performantelor aplicatiilor. Avantajele difera , functie de mediul in care PL/SQL este folosit. Aceste avantaje vor fi discutate mai tirziu in acelasi unit.

----- Desene pag 19-4 -----

- Structura PL/SQL
Fiecare unit al PL/SQL contine unul sau mai multe blocuri. Aceste blocuri pot fi complet separate sau lipite unul de altul. Astfel, un bloc poate fi numai o mica parte a altui bloc, care la rindul lui poate fi numai o parte din intregul program.

De obicei, un bloc poate fi un bloc anonim sau un sub-program.

- Blocuri anonime

Uzual sunt blocurile fara un nume. Aceste blocuri sunt declarate in punctul din aplicatie unde vor fi folosite, si transferate modulului PL/SQL la momentul executiei. Blocurile anonime pot apare in cadrul unui program precompilat (sau program OCI) si cu SQL*Plus sau SQL*DBA. Partile de executie din SQL*Forms contin asemenea blocuri.

- Sub-programe

Acestea sunt denumite blocuri PL/SQL. Pot fi declarate ca proceduri sau ca functii. In ultimul caz, ele intorc o valoare daca sunt apelate. SQL*Forms permite ca procedurile si functiile sa fie declarate ca parte a Form-ului, si apelate din orice punct.

In acest curs ne vom concentra mai mult asupra blocurilor anonime. Definirea unui bloc anonim este urmatoarea:

```

-----
|          DECLARE          |
|          Definitia obiectelor PL/SQL care vor |
|          fi folosite in acest bloc.          |
-----
|          BEGIN          |
|          Actiuni executabile          |
-----
|          EXCEPTION          |
|          Tratarea erorilor (cazuri de exceptie) |
|          ce trebuie facut daca o actiune cauzeaza |
|          o eroare          |
|          END;          |
-----

```

Cuvintele cheie BEGIN si END sunt obligatorii si delimiteaza corpul actiunii ce urmeaza a fi executate. Sectiunea DECLARE este optionala si se foloseste pentru a defini obiecte PL/SQL cum ar fi variabile folosite in cadrul blocului sau un bloc inclus in acesta. La sfirsit sectiune EXCEPTION este folosita pentru a minui erori predefinite, si defineste actiunile ce vor fi intreprinse daca astfel de conditii apar. De asemenea, sectiunea EXCEPTION este optionala, si este plasata imediat inainte de clauza 'END'.

Sa urmarim un exemplu de bloc, si sa fim atenti la structura si tipul actiunilor :

```

DECLARE
  qty_on_hand    NUMBER(5);
BEGIN
  SELECT quantity INTO qty_on_hand FROM inventory
  WHERE product = 'TENNIS RACKET';

  IF qty_on_hand > 0 THEN
    UPDATE inventory SET quantity =
quantity -1
    WHERE product = 'TENNIS
RACKET';

    INSERT INTO purchase_record
VALUES ('Tennis Racket
purchased', SYSDATE);
  ELSE
    INSERT INTO purchased_record
VALUES ('OUT of Tennis
Rackets', SYSDATE);
  END IF;
COMMIT;

```

```

EXCEPTION
  WHEN no_data_found THEN
                                INSERT INTO error_table
                                VALUES ('Product Tennis Rackets
not found');
END;

```

Retineti ca sectiunea de cuvinte cheie DECLARE, BEGIN, EXCEPTION nu sunt urmate de ';' dar END si toate celelalte instructiuni PL/SQL au nevoie de ';'. Acest lucru permite sa scriem mai multe instructiuni pe aceiasi linie daca avem nevoie.

Exemplul anterior daclara o variabila PL/SQL qty_on_hand si apoi foloseste instructiunea SELECT pentru a completa cu Tennis Rackets. Tabela 'inventory' este apoi actualizata daca valoarea este pozitiva.

Daca instructiunea SELECT esueaza in a gasi o linia cu acest produs, atunci se genereaza eroarea 'no_data_found' si controlul este pasat imediat sectiunii EXCEPTION unde problema este inregistrata in tabela de erori.

- Mediul de lucru PL/SQL
PL/SQL nu este in sine un produs ORACLE. Este o tehnologie folosita de RDBMS si de anumite aplicatii ORACLE. Blocuri ale PL/SQL sunt pasate de asemenea si procesate de unitatea de executie PL/SQL. Unitatea folosita va depinde de locul in care PL/SQL este apelat.
- Unitatea de executie PL/SQL
Aceasta unitate executa instructiuni procedurale, dar trimite instructiuni SQL catre Procesorul de Instructiuni SQL in cadrul ORACLE RDBMS. In acest fel unitatea de executie filtreaza instructiunile SQL, si instructiunile procedurale ramase sunt procesate in Unitatea de Executie A Instructiunilor Procedurale, care este in interiorul unitatii de executie.
- PL/SQL in programe SQL*Plus si SQL*DBA
Blocurile care sunt transmise din acest mediu sunt procesate de catre unitatea de executie in RDBMS.SQL din cadrul blocului nu este spart in instructiuni separate pina cind ajunge la unitatea PL/SQL, unde comenzile SQL sunt trimise unitatii de executie SQL. Acest lucru inseamna ca un singur transfer este necesar pentru a trimite blocul de la aplicatie la ORACLE, imbunatatind performantele in mod special intr-o retea de tipul client-server. ----- Poza de la pag. 19-7 -----

Folosirea PL/SQL in SQL*Plus este discutata in Unitul 2. Folosirea PL/SQL in alte medii care folosesc unitatea RDBMS este subiect pentru viitoare cursuri.

- PL/SQL in SQL*Forms V3, SQL*Menu V5 si Aplicatii CDE Oracle
Multe aplicatii ORACLE, incluzind partea din CDE (Medii de Dezvoltare

Cooperante) au propriile unitati de PL/SQL, care sunt independente de orice alta unitate prezenta in RDBMS. Asa cum unitatile rezida in aplicatii comenzile SQL sunt separate inainte de a fi transferate catre ORACLE.

Procesarea datelor locale aplicatiilor (care este in interiorul mediului de lucru client, spre deosebire de baza de date) poate fi procesata de Procesorul de Instructiuni PL/SQL, fata de folosirea SQL, deci reducind munca trimisa catre RDBMS, si numarul de locatii de memorie necesare. ----- Poza 19-8 ----

Retineti ca Procedurile si Functiile declarate ca parte a aplicatiilor CDE si SQL*Forms sunt diferite de acelea care pot fi stocate in baza de date cu optiunea Procedural, altfel structura generala ramainind aceiasi.

- Facilitati PL/SQL
Urmatoarea parte a unit-ului discuta avantajele (facilitatile) ale PL/SQL comune fiecarui mediu de lucru unde limbajul este instalat.

Acesta este un sumar.

- Variabile si Constante

pentru memorarea si manipularea valorilor. Tipurile de date includ NUMBER, CHAR si DATE.

- SQL suport

SELECT, INSERT, UPDATE, DELETE, COMMIT, SAVEPOINT si ROLLBACK sunt instructiuni suportate.

- Tipuri de date compuse

Recordurile permit definirea grupurilor de cimpuri si folosirea in blocuri PL/SQL. Versiunea 2 suporta de asemenea tabele, la fel ca sirurile.

- Controlul Executiei

Instructiunea IF, Bucle, salturi (GOTO) si etichete ofera actiuni conditionale, teste complexe, bucle iterative.

- Functii Interne

Multe functii SQL sunt disponibile pentru a manipula variabile si valori PL/SQL.

- Tratarea Exceptiilor

Avertismentele si Erorile sunt defnita ca exceptii, care pot fi minuite in fiecare bloc. Exceptiile defnite de user pot fi declarate si folosite la fel de bine ca erorile standard ORACLE.

- Stocarea de cod in baza de date

Optiunea Procedural din ORACLE suporta functii si proceduri care pot fi memorate in baza de date si facute disponibile pentru toate aplicatiile.

□